Basic Web Application Development

Spring 2015 – 1 credit hour
Student Taught (Satisfactory/Unsatisfactory)

Matthew Schurr
mschurr@rice.edu
(404) 395-0511

Instructor

Matthew Schurr

Duncan College

Class of 2016

Phone*: (404) 395-0511

* If you are going to contact me by phone, send me a text message with your name before calling.

Email: mschurr@rice.edu

Faculty Sponsor

Professor Scott Rixner

Department of Computer Science

Email: rixner@rice.edu

Course Objectives and Expectations

This course will explore topics in web-based application development and deployment using the latest industry-standard technologies. By the end of the course, students will have...

- a fundamental understanding of how web-based applications function
- a fundamental understanding of the Model-View-Controller (MVC) software architecture
- a fundamental understanding of the following technologies: HTTP, HTML, CSS, SQL, and PHP
- the knowledge and tools necessary to build their own web applications
- a strong basis in web application development that makes them feel confident enough to explore more advanced topics on their own through other university courses or through freely available internet resources

At the end of the semester, students will build a functioning version of the Duncan College Book Exchange (with the guidance of the instructor) to reinforce all of the information they have learned.

This course will require between twenty minutes and two hours of homework each week depending on your skill level. The instructor expects that you will complete all of your assignments because you are interested in learning a fun and useful skill that will make you more attractive to employers.

Prerequisites

You should read this very carefully before enrolling in the course. If you do not meet these requirements, you will probably struggle to complete your assignments within a reasonable amount of time.

Due to the short length of classes, we will not be able to cover many basic programming concepts. You will also be expected to learn a new programming language and several markup languages in a single semester.

In order to enroll in this course, you should meet the following requirements:

- You intend to major in Computer Science or Electrical Engineering.
- You are not a graduate student.
- You have taken COMP 182 Algorithmic Thinking and COMP 215 Intro to Program Design **OR** you have significant programming experience (in any general-purpose language – not Matlab) and are confident in your ability to pick up new programming concepts/languages quickly.

If you are not sure whether or not you should take this course, contact the instructor.

Grading Policy

Your final grade will be based on the assignments you turn in, your attendance, and a final project.

Assignments are worth 60% of your total grade. Assignments are primarily completion grades; as long as you make an effort to complete the assignment, you will receive full credit. Each assignment is weighted equally. After the due date for each assignment, the instructor will post solutions. You should read through these and be sure that you understand them. No credit will be given for late assignments.

Attendance is worth 20% of your total grade. Each class day is weighted evenly. You will receive two excused absences.

The final project will be graded by the instructor for accuracy and is worth 20% of your total grade. If you turn in the project late, you will lose 10% of your project grade for each day that the project is late.

At the end of the semester, students who have earned a grade of 60% or more will receive a passing grade.

Absence Policy

Attendance is worth 20% of your total grade. You will lose points on days that you do not show up to class. You will be forgiven two excused absences.

Enrollment

This course will have an enrollment cap of 19 students.

Office Hours

Office Hours for this course will be by appointment only. If you require help, please contact the instructor by email to establish a meeting time and location.

Required Texts and Materials

There are no required texts for this class. However, you are expected to have access to a computer.

The following free software is recommended for the class:

- Firefox (http://getfirefox.com/) or Chrome (https://www.google.com/chrome/)
- Sublime Text (http://www.sublimetext.com/)
- Virtual Box (https://www.virtualbox.org/)

Course Website

The course website can be found on Owl Space. You will turn in your assignments and view your grades using Owl Space. The instructor will also post lecture notes and resources on Owl Space.

Meeting Times

We will meet once a week on Thursdays from 7:00 PM – 8:30 PM in TBA.

Papers and Examinations

You will not have any long essays or examinations in this class.

Honor Code Policy and Expectations

I am teaching this class using a practical approach. That means that I encourage you to collaborate with your fellow students and use any available Internet resources, just as you would at a real-world job. As such, you may freely share segments of code with your fellow students, but you **may not** give them the entire or majority of the solution to an assignment. You can also copy code that you find on the Internet. If you do share or copy code, you must provide a comment that indicates where you got it.

Important: In the end, copying code without understanding it does nothing for you. The goal is for you to learn how to make your own web application. Therefore, I expect you to understand how any third party code that you use works (regardless of its source) and be able to fully explain its operation to me (the instructor). If you

are unable to explain how or why something works, you may not use it in your project or assignments (unless the instructor gave it to you).

Special Accommodations

Any student with a disability that requires accommodation is encouraged to contact both the course instructor and University's Disability Support Services.

Syllabus Changes

The information contained in this syllabus, other than absence policies, may be subject to change with reasonable advance notice, as deemed appropriate by the instructor.

Class Schedule

This class consists of five units. Each unit will be accompanied by a homework assignment. The course will get progressively more difficult over the course of the semester; as a result, significantly more time is allocated to the later units and assignments.

UNIT 1 – The World Wide Web (Week 1)
In order to create web applications, you must first understand the foundation upon which they are built. This unit will explain the basic components of the Internet and how they interact to enable the creation of complex, interactive applications. You will also learn the fundamentals of the Model-View-Controller (MVC) software design pattern.

Assignment 1: You will be observing the HTTP protocol in action and summarizing your observations.

UNIT 2 – Hypertext Markup Language (Weeks 2-3)
We are now ready to explore the first part of our MVC framework: the View. The most fundamental part of any application is the ability to provide users with visual feedback. Websites define their user interface using the HyperText Markup Language (HTML). In this unit, we will learn how to format HTML documents and create interactive forms that can be filled out in your browser.

Assignment 2: You will be creating a basic quiz on a topic of your choice that displays in your browser.

UNIT 3 – Cascading Style Sheets (Weeks 4-6)
We now know enough to create a basic (but fully functional) user interface in HTML. However, almost all modern Internet users expect colorful, feature-filled visual displays that are very different from the simple ones you have learned to create. In this unit, we will learn how to create these more advanced layouts and user interfaces by using the Cascading Style Sheet language to enhance our HTML documents.

Assignment 3: You will be designing a basic layout for a book exchange web application that displays in your browser.

UNIT 4 – Remote Procedure Calls, Relational Database Systems, SQL (Weeks 7-9)

***Guest Video Lecturer*:** Professor Jennifer Widom (Stanford University)

Now that we know how to create a graphical user interface (GUI) for our web application, we are ready to explore the next part of our MVC framework: the Model. Put simply, the model is the format in which we store our application's data. The model bundles together our data and the algorithms that act on it into one convenient package.

Assignment 4: You will complete a series of exercises from Stanford University's Introduction to Database course and create a database schema for a book exchange web application.

UNIT 5 – Server-Side Programming in PHP (Weeks 10-14)

We are now ready to explore the last part of our MVC architecture: the Controller. The controller is the glue that combines the Model, View, and HTTP together to create a coherent, interactive application.

Assignment 5: You will complete a series of exercises designed to introduce you to the PHP programming language, interacting with databases programmatically, interacting with HTTP requests, dealing with sessions, and rendering HTML content dynamically. You will demonstrate your knowledge by implementing a grading system for the test you created in Assignment 2.

Final Project (Weeks 12-14)

Your task is to complete the book exchange that we have been working on in class. You have been provided with significant help – most of the files you require have been created for you with comments that will guide you through the process, and a large part of the work will be completed in class. You have also been provided with libraries that handle sessions and user authentication. Your application should have the same basic functionality as the Duncan College Book Exchange and you should use it as a reference when designing yours. If you wish to add additional components or functionality, you may choose to do so, however you will only be graded based on the criteria below.